

Introduction

We can't keep adding processors or electrical power for running scientific applications on supercomputers [1]. One possible solution is lower precision computing.

Background

Scientific computation can benefit from lower precision computing. The IEEE 754 standard defines floating point numbers that take 32 or 64 bits of storage, known as **single** and **double** precision respectively. The standard also defines a floating point format for 16 bits of storage, known as **half** precision. Half precision exists on GPUs and may be a codesign opportunity on CPUs as well.

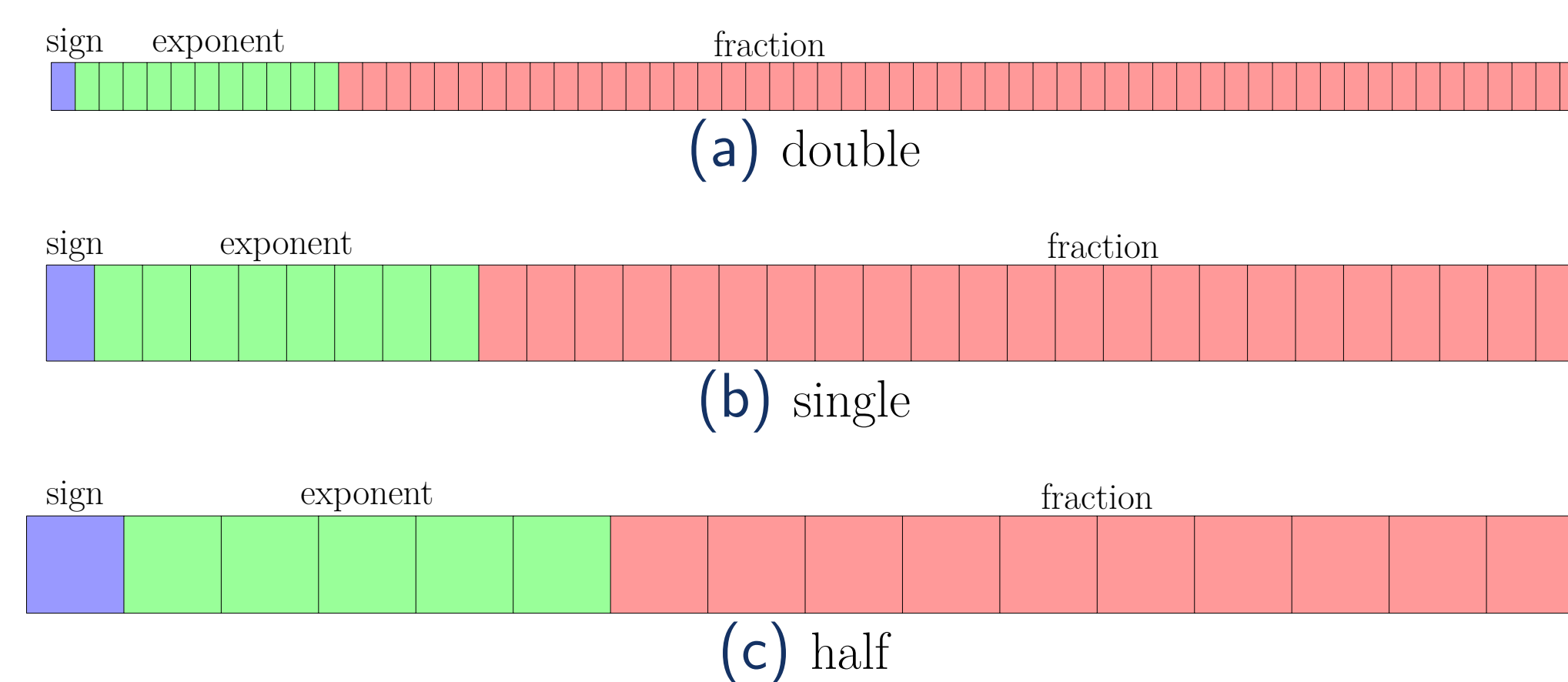


Figure 1: Different Precisions, IEEE 754 standard

Lower precision benefits

- More efficient use of memory bandwidth.
- Halves the memory footprint.
- Increase number of vector operations with current AVX architecture.

Fogerty et al. [2] showed two examples of applications that benefit from lower precision methods and experimented on a variety of architectures. We continue their work by using mixed precision on a different application, Tycho2, a LANL mini-app.

Application: Tycho2

Solves a key kernel in radiation transport codes [3].

- Latency, bandwidth, and computationally bound.
- 6-dimensional arrays take up most of the computation:
 - Q (source)
 - Q_{total} (intermediate calculation)
 - Ψ (solution).
- Unstructured mesh causes problem to be latency bound.

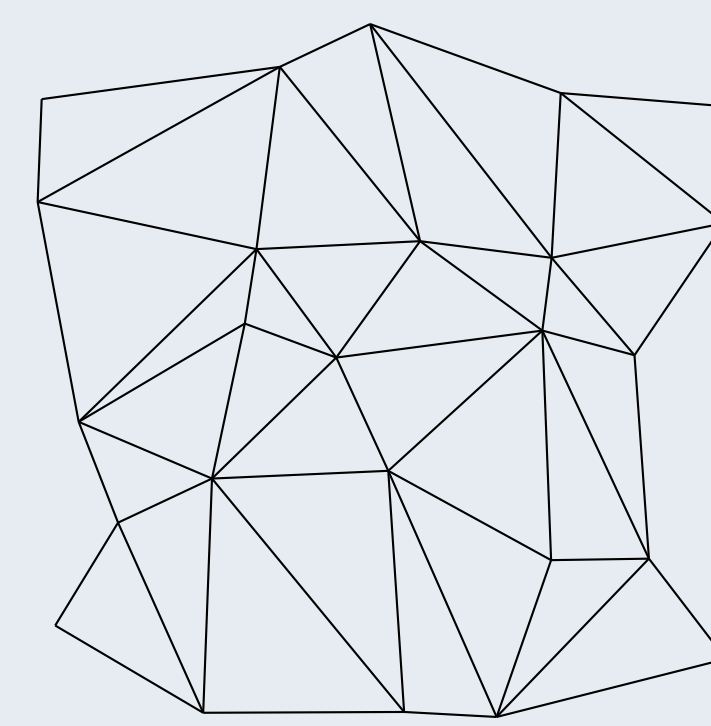


Figure 2: Representing Tycho2 unstructured tetrahedral mesh. Drawing is 2D but actual mesh is 3D.

Methods

Original code had Q , Q_{total} , and Ψ in double precision. We created two ports as shown in table 1.

Q	Q_{total}	Ψ
double	double	double
single	single	single
single	double	double

Table 1: Original version is all double precision. We use all single precision as well as a mixed precision method to see how the trade-offs affect computational accuracy and runtime.

We used templates to toggle Q between single and double precision. For scaling, we test 32-way parallelized versions of the code, with 32 MPI processes per node with 1 thread per process.

Fixed Budget Computation

It is also possible to keep memory or energy use constant. We may run at a lower precision at a higher resolution (vice a high precision at low resolution) and get an equivalent fidelity with the same memory budget.

Future Work

By showing that lower floating point precision is acceptable for scientific research, hardware designers may take note and design GPUs to account for half-precision floating point operations as well. Different directions the research can take:

- What levels of precision still provide a correct answer?
- How can the error of the solution be bounded with respect to the available levels of precision?
- Can we specify the lengths of the mantissa and exponent for specific applications? What about non-powers of two?
- Are some applications faster and more accurate using fixed point rather than floating point?

Results

Single precision gives solutions that are visually indistinguishable from double, but has faster runtime.

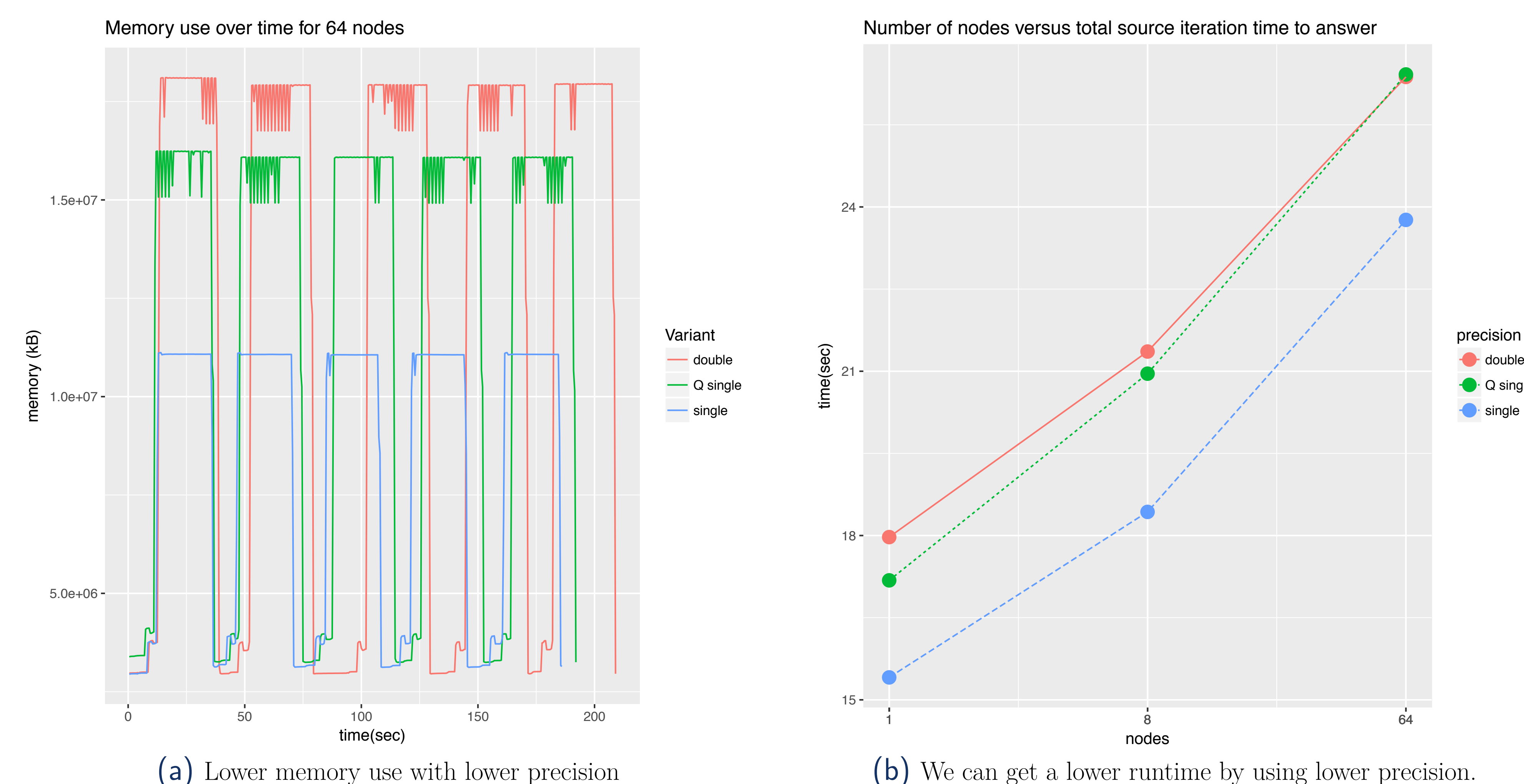


Figure 3: We can get lower cost for similar errors.

References

- ASCAC Subcommittee et al. Top ten exascale research challenges. *US Department Of Energy Report*, 2014.
- Shane Fogerty, Siddhartha Bishnu, Yuliana Zamora, Laura Monroe, Steve Poole, Michael Lam, Joe Schoonover, and Robert Robey. Thoughtful precision in mini-apps. In *Cluster Computing (CLUSTER), 2017 IEEE International Conference on*, pages 858–865. IEEE, 2017.
- Tycho2. <https://github.com/lanl/tycho2>.

Acknowledgements

Many thanks to our mentors: Laura Monroe (HPC-DES), Bob Robey (XCP-2), Kris Garrett (CCS-2), and Hai Ah Nam (CCS-2). Support for this project was provided by U.S. Department of Energy at Los Alamos National Laboratory supported by Contract No. DE-AC52-06NA25396. Data for this project was collected on the Darwin cluster at Los Alamos National Laboratory, and the Cori cluster at the National Energy Research Scientific Computing Center.